Philosophical Studies

Varieties of Biased Algorithms

--Manuscript Draft--

Manuscript Number:	PHIL-D-21-00991
Full Title:	Varieties of Biased Algorithms
Article Type:	S.I. : Pacific APA 2020 & 2021
Keywords:	AI; machine learning; algorithmic bias; top-down biases; bottom-up biases; implicit bias

Abstract:	We are living in the age of AI. Algorithms are used to make decisions about criminal sentencing, loans, policing, credit card applications, job recruiting, and medical care. We don't just use the fruits of AI systems. Our actions provide the data on which they operate, and they are being used to make decisions about us. Contrary to what is frequently assumed both among the public and researchers, most algorithmic biases stem not from the effects of a programmer's beliefs, goals, and background views, but rather from how AI systems function at the lowest level: the patterns that the algorithm detects within the data it receives, the connections it forges between data points, and the generalizations that emerge from these connections. They are bottom-up biases rather than top-down biases. While top-down biases are due to the goals, expectations, or beliefs of the programmer designing the algorithm, bottom-up biases are due to incoming data and its processing at the lowest level. This paper analyzes several sources of such bottom-up biases and investigates how these sources interact.

Varieties of Biased Algorithms

We are living in the age of Al. Algorithms are used to make decisions about criminal sentencing, loans, policing, credit card applications, job recruiting, and medical care. What news we see on social media and which ads we are shown online are determined by algorithms. We do not only use the fruits of AI systems. Our actions provide the data on which they operate. Based on this data, algorithms are being used to make decisions about us that range from the superficial to the life changing. Since the pandemic, our lives have moved online to an unprecedented degree. Consequently, we are ever more subject to the positive and negative consequences of AI.

All AI systems have biases. Many of these biases are deeply harmful especially for already marginalized demographic groups. To give a concrete example: If one does on online search for a name predominantly given to black babies—say, Deshawn, Aisha, Jermaine—one is more likely to get an ad for criminal background checks than if one searches a name predominantly given to white babies—say, Geoffrey, Jill, or Emma. I will come back to this example throughout this paper. Here are just a few further examples of bias in AI. There are racial biases in the health care algorithms used in hospitals throughout the United States as well as in recidivism risk software used in courtrooms across the United States to make judgments about bail. There are biases in hiring algorithms used to filter out promising job applications at companies. Recently, it transpired that the algorithm used by Apple Card to determine the creditworthiness of applicants seems to set the credit line for men significantly higher than for women even when the women have a better credit score.

In addition to well-known and well-studied top-down biases, AI is riddled with bottom-up biases. To a first approximation top-down biases stem from the beliefs and world views of programmers, whereas bottom-up biases stem from incoming data and its processing at the lowest levels of AI systems. Bottom-up biases are rarely recognized and barely understood. The aim of this paper is to create a road map for understanding the characteristics and mechanisms of bottom-up biases.

I will proceed as follows. First, I introduce the distinction between top-down and bottom-up biases in more detail (Section 1). I argue that bottom-up biases occur not only in AI but equally in cognitive and perceptual systems (Section 2). Then I distinguish a range of stages and levels at which bottom-up biases occur (Section 3). This allows me to distinguish three types of bottom-up biases: Fine-Tuning Bias, Feature-Linking Bias, and Training Sample Bias (Section 4).

The aim is not to minimize the effects of top-down biases: they are rampant. Rather I am shedding light on a different kind of bias that is rarely recognized and the nature of which is barely understood. It is important to recognize the difference between top-down and bottom-up biases since the strategies for mitigating them are very different. While I will not here discuss ways to mitigate

algorithmic bias, I hope that a clear analysis of their nature will set the stage for productive mitigation strategies.

Before I embark on this project, it is necessary to make a terminological point. Colloquially as well as in the humanities and social sciences, the term "bias" is typically used to refer only to harmful biases. In the sciences, however, the term is standardly used to refer to any partiality or systematic weighting of information, be it harmful or beneficial. I will be using the term in this latter sense. The reason for this will soon become clear.

1. Top-Down and Bottom-Up Biases

Biases are typically discussed in the framework of top-down biases, that is, biases that stem from the effects that a person's beliefs, concepts, and background views have on her perceptions, thoughts, and actions (Brownstein and Saul 2016). If someone has racist or sexist beliefs, this will likely affect how she interacts with people. Such top-down biases can be conscious or unconscious, explicit or implicit. A programmer with such beliefs may design AI systems that exhibit the racist or sexist beliefs she harbors. Moreover, her background beliefs may affect how she interprets the outputs generated by an algorithm. There is ample evidence that such top-down biases exist in our perceptual and cognitive systems and that such top-down biases can affect how AI systems are designed and how their outputs are interpreted.

In addition to being effected by top-down biases, AI systems are riddled with bottom biases, that is, biases that stem from how AI systems function at the lowest level: AI systems operate on colossal and complex datasets. The datasets consist of pieces of information, which we can call features. Algorithms detect patterns within the data it receives, pick up on regularities and correlations between features within the data, forge connections between these features, generalize from these connections, and then make predictions about the future. So, they are not top-down, but rather bottom-up biases. Indeed, the correlations between features may be so complex as to be beyond human understanding even by the programmers who produced the initial algorithms. Yet they are not beyond algorithmic discovery.

While top-down biases are due to the conscious or unconscious, explicit or implicit goals, expectations, or beliefs of the programmer designing the algorithm or assessing its outputs, bottom-up biases are due to incoming data and its processing at the lowest level.

To illustrate the difference between these two kinds of biases, let's consider again the ads one gets when one searches certain names. No programmer deviously wrote code so that ads for criminal background checks were prompted when a name typical given to black babies had been searched online. What happened is perhaps even more worrisome.

The bias came about due to the data on which the algorithm operated. The data was provided

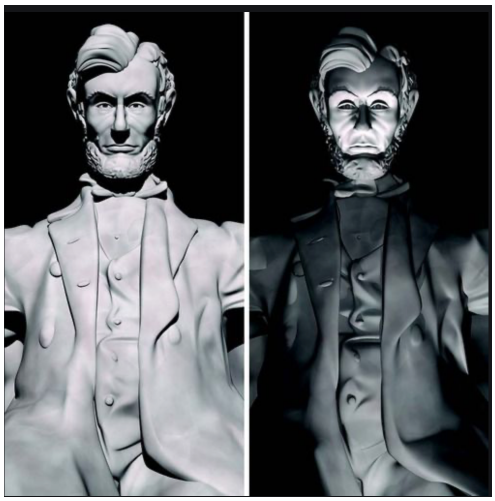
by us: the millions of people using google each day across the globe. Enough of us must have done criminal background checks on certain kinds of names shortly after having searched for those names online. AdSense, the google ad algorithm, picked up on this correlation, linked features, and delivered ads suggestive of an arrest record in connection to the search of some names but not others. This is a classic case of a bottom-up bias.

2. Bottom-Up Biases in Natural and Artificial Intelligent Systems

Why are AI systems and machine learning algorithms riddled with bottom-up biases? The reason is that there is a mismatch between processing power and the quantity of data that the system is tasked to process. To deal with this mismatch, such systems take short cuts. More specifically, they make statistical generalizations since the data on which they operate far outstrips their processing power. So they use statistical generalization to operate efficiently.

Whenever there is too much information with too little processing power, there will be biases. Our perceptual and cognitive systems face this same problem. Due to having to limited processing resources and the mismatch between their processing power and the quantity of data that the system is tasked to process, they have bottom-up biases.

The human visual system, for example, is pervaded with bottom-up biases. Some of these biases are beneficial. We have depth biases that allow us to recognize distances more efficiently. We have a horizontal bias. The visual systems of most mammals have the bias that light comes from above: the so-called top-down lighting bias. It is beneficial since light typically comes from above. The bias allows the visual system to process incoming information more efficiently. It allows us to recognize the statue as depicting Lincoln more quickly when it is illuminated by light coming from above, than when it is illuminated from below.



Occasionally, the bias leads to error, such as when we erroneously see something to be concave that is in fact convex and illuminated from below. Despite occasionally generating mistakes, such biases are, however, beneficial in that they make the visual system more efficient.

Other biases are harmful. One striking case of a harmful bias in our visual system is the so-called cross-race bias, which is the tendency to more easily recognize faces of the ethnic group that one is most familiar with (which is often, but not necessarily, one's own ethnic group). All else being equal, individuals of a given ethnic group are distinguishable from each other in proportion to their familiarity with people of that ethnic group. Cross-race bias is universal. There are numerous studies done with people of different ethnic backgrounds. There are studies done that compare Hispanic with black and white participants, black with white and Japanese participants, Chinese with Indian and Korean participants.[1]

One standard explanation of the cross-race bias is what is called the perceptual expertise hypothesis. In a nutshell, the idea is that people develop more fine-tuned discriminatory capacities in distinguishing between faces of ethnic group members of which they interact with most. Studies show that the more a person is exposed to faces of an ethnic group she has not previously interacted with much, the better she becomes at recognizing faces of that ethnic groups. So with exposure the cross-race bias falls away. That fact supports the perceptual expertise hypothesis. The fact that mere exposure helps mitigate the bias is evidence that the cross-race bias is at least in part a bottom-up bias.

Cross-race bias is problematic for many reasons: the accuracy of eyewitness memory is significantly affected by the ethnic identity of both the suspect and the eye-witness. Most individuals more accurately recognize a face belonging to their ethnic group than an individual whose ethnic group differs from that of their own. Moreover, a meta-analysis of several studies about emotion recognition in facial expressions has revealed that people can recognize and interpret the emotional facial expression of a person of their own ethnic group faster and better than of a person of a different ethnic group (Shapiro et al. 2009; Brooks and Freeman 2017). There are many similar such biases. For example, studies show that there is an own-age bias where people are better at recognizing people of a similar age as themselves (see Hills and Lewis 2011). Visual systems should be reliable, efficient, and accurate. Beneficial biases help make it reliable and efficient. But a visual system should be free of harmful biases.

Given how flawed humans are, it may not be surprising that we are biased. It may come as a surprise, however, that AI systems are biased. After all, computer algorithms form the core of AI systems and these algorithms are grounded in mathematics and operate on raw data. So one might expect them to be objective and just. Unfortunately, they are not.

As in our perceptual and cognitive systems, not all short cuts that machine learning algorithms

make are problematic. Some are unproblematic and even beneficial. The task is to get rid of the problematic ones while retaining the beneficial ones.

Before I proceed to addressing bottom biases in more details it is worth addressing an objection waiting in the wings. You might be concerned that I am discussing human and artificial intelligence in the same breath. While there are deep differences between the two, they are similar in two aspects that are central for our purposes here. I will use human vision as an example, but the points generalize to other aspects of our minds. As machine learning algorithms need to be trained on data, our visual system develops by interacting with its environment. Moreover, our visual system is algorithm based. Of course, it is not formalized in written code, but it operates according to similar algorithmic principles: it operates on input information, processes that information (sometimes with priors), makes connections between information, and it has an output, namely, perceptual states.

It is a hotly debated question just how similar machine learning algorithms are to human visual and cognitive systems. Two critical differences are that AI systems are not embodied and that they need significantly more data than humans to gain the same learning effect.

Despite these difference, there are the two similarities mentioned. These two similarities are sufficient for our purposes. Putting aside differences in implementation, there are reasons to think that the bottom-up biases in AI systems and our visual and cognitive systems operate in similar ways. This is not a surprise given that AI mimics human learning. I will argue that all the bottom-up biases that I will be distinguishing within AI systems occur in our visual and cognitive systems as well.

3. Machine Learning Algorithms

To discuss bottom-up biases in more detail, we need to take a closer look at what algorithms are. At its most basic level an algorithm is a series of instructions for performing a specific task. In a machine learning algorithm, many steps of the algorithm are not written down in code by a programmer. Instead the algorithm is derived from input data. A programmer might be involved in coding the process by which the final algorithm or model is derived from the data, but she does not design the model itself.

Machine learning is a form of self-programming, since the data itself determines the details of the model. Such data-driven self-programming allow AI to mimic human learning and produce algorithms for complex tasks such as language recognition, face recognition, language translation, and other such tasks at which humans excel. From the point of view of machine learning, such tasks are all thought of as prediction tasks.

Data processing in the visual system and in AI systems abides by rules, but neither the rules nor

the stages of the transformations need to be represented.[2] Indeed, AI systems standardly operate in a purely connectionist framework with no representations in play at the stage of data processing. The algorithm itself is not represented but rather encoded in the system. There are reasons to think that the visual system operates in similar fashion on the basis of non-representational transformation principles (Burge 2010, p. 424). More generally, we can say that there are two ways for data processing to include representations: one is for it to operate on representations, the other is for the rules governing the processing to be represented. Data processing in the visual system and AI system need not be representational in either way. Of course, in our visual or cognitive systems biased processing may include implicitly held beliefs. But it need not. Biased processing may simply involve linking two features where neither the features nor the link between them need to be representation.

There are three basic types of machine learning programs[3]:

- supervised learning
- unsupervised learning
- reinforcement learning

For the sake of concreteness, I will focus on supervised learning programs. But everything I say about those generalizes, with some simple tweaks, to the other two forms of machine learning. After the setup, which includes:

- task definition
- dataset construction
- model definition

there are three main phases of supervised learning:

- training phase
- test phase

During the *training phase*, the program is trained on already-classified data, that is, the training sample. This allows the program to learn the relationship between features and classifiers. These classifiers are frequently referred to as labels. It is important to note that the labels (or classifiers) are not names, words, or anything symbolic. They are simple markers that discriminate data into two or more groups, thereby classifying them. The algorithm identifies patterns in this known dataset and produces a feature space and a predictive model dedicated to replicating those patterns for new data. As I will discuss in

[2] Traditionally, biases—be they implicit or explicit—have been understood as based on unconscious beliefs. For defenders of such representational views, see De Houwer (2014), Levy (2015), Mandelbaum (2015), and Carruthers (2017). For a criticism of such views and a defense of an alternative functionalist characterization of bias, see Johnson (forthcoming-a). The argument of my paper is neutral on whether biases are representational or functional. That said, the biases in AI systems are for the most part not representational and empirical evidence suggests overwhelmingly that biased processing of our visual system is at least in part not representational. For discussion, see Johnson (forthcoming-b).

[3] Humans can learn according to each of these three types as well.

more detail shortly, if there are problematic patterns encoded in the training data, then its predictions will have those same problematic patterns. This is the key source of training-sample bias.

In the *test phase*, the algorithm gets applied to unclassified data—test data—and the algorithm classifies the data within the feature space developed during the training phase. More specifically, it classifies new data on the basis of its proximity to known classifications in the feature space. An important fact is that for any task, even trivial ones, such as sorting, there can be multiple alternative algorithms with different strengths and weaknesses.

After the test phase, the algorithm is released into the wild, so to speak. During this *deployment phase* the algorithm continues to learn in an unsupervised fashion. It continuously builds and expands its network, thereby linking with new features and adjusting the links formed during the test and training phase. Thereby the neural network identifies new patterns and uncovers structure among the data on which it operates.

4. Biased Input, Biased Model Selection, Biased Networks, Biased Output

It is still widely believed that algorithmic is a biased data problem.[4] This stance exonerates programmers and minimizes what they can do to avoid producing biased machine learning programs. More importantly it is simply false. Bias can occur at every stage of an algorithm: it can occur in designing the initial algorithm, at the data input stage, the model selection stage, the processing stage which includes which features are linked and how those links are weighted, and the output stage. Moreover, bias can occur when the output is interpreted. So we can distinguish seven stages at which bias can occur.

- Design bias
- Biased input (at training, testing, or deployment stage)
- Model selection bias
- Biased network
- Biased output
- Biased interpretations

What we ultimately care about is that the output of an algorithm is not biased. However, output bias is a result of bias at one or more of the five previous stages of the algorithm. In order to avoid a biased output, we need to make sure that there are no biases at those stages. Even if the output is free of bias, it can be interpreted in a biased manner. So vigilance is required every step of the way.

[4] For recent discussion, see Hooker 2021.

Before we discuss each of these stages in more detail, it is important to note that some of these biases are pure top-down biases, such as design bias, biased input at the training phase, and biased interpretations of the output of algorithms. The rest are either pure bottom-up biases or a mixture of top-down and bottom-up biases. Since design bias, model selection bias, and biased interpretations are pure top-down biases, I will address them only briefly here. Bottom-up biases are barely discussed, I will here focus on those. So I will focus on the biases at the input and processing stages of an algorithm. It needs to be noted that top-down biases can occur at these stages as well. But as I will argue many are bottom-up biases.

At the design stage, the hypothesis space is set up, the objectives of the algorithm are defined and it is decided how they are prioritized. Any biases that occur at this stage are pure top-down biases.

A critical part of setting up a machine learning algorithm is to select a model from a multitude of candidate machine learning models. A choice needs to be made as to which model will perform best in making predictions for the problem at hand. All models have some predictive error, given the statistical noise in the data, the incompleteness of the data sample, and the limitations of each different model type. Model selection is a key source of algorithmic bias. It is a source of bias that stems entirely from the decision making of process of programmers. Thus, model selection bias is a top-down bias. Most machine learning algorithms include an optimization algorithm that ensures the optimal weighting of the parameters of the algorithm. I include this as part of the model selection and like model selection bias it is a pure top-down bias.

The selected model will then be trained on a data set. There are different stages of data input: data fed to the algorithm during the training phase, test phase data, and the data to which the algorithm is exposed during the deployment phase. At each of these phases, the input on which an algorithm operates can be biased. If there is human bias in the data, the algorithm will replicate and possibly amplify this bias.

In human cognition or perception, any base on which we perform actions or form belief can be a biased input. The relevant base could be information from our environment, our world view, a concept, or a mental state, such as a biased belief, attitude, or question. A discriminatory judgment formed on a biased base, can itself be the biased base of further discriminatory judgments or actions. Similarly, the output of an algorithm can itself be the biased base of further operations of that algorithm.[5]

During the training, testing, and deployment phase the AI system continuously expands its network of nodes and weighted links between features. I will refer to such structures of nodes and weighted links as a *network*. In neural networks this structure of nodes and weighted links is called an

[5] As I argue in Section 5, training sample bias is a case of biased input at the training phase.

architecture. In Bayesian networks it is called a graph. It can also be called a tree or simply a network. A model is the mathematical representation of a network and a network in turn is an instance of a model.

Network bias occurs if features that co-occur in our environment are linked even though there is nothing robust that links them other than, for example, a history of discrimination. More generally, biases can occur at this stage in virtue of which features are classified, which features are linked, and how those links are weighted. Network bias can be a result of biased input, biased model selection, or any number of other factors.[6]

Biases at any of the previous stages will typically lead to biased outputs of the AI system. These outcomes can take many forms. They can be biased verdicts, assessments, or biased products. In the human case, such biased outputs may be biased actions or beliefs that we arrive at due to biased input or biased processing.

It is worth stressing that a biased output of an AI system can form the basis for further discriminatory judgments or actions. If it does, it serves as a biased input for further operations of that system or other systems. This is important since this is one central way in which bias feeds bias. Similarly, in humans a biased belief can at different stages be both a biased output and a biased input.

Bias can occur not only at every stage of an algorithm, but moreover at every level at which information is processed: the computational, the neural, the connectionist, and the representational level.[7] There is a lot to be said about how bias occurs at each of these levels, but I will not dwell on this here.[8]

5. Three Kinds of Bottom-Up Biases: Feature-Linking Bias, Fine-Tuning Bias, and Training Sample Bias

In light of these distinctions between different stages and levels at which bottom-up biases can occur, we are in a position to gain a clearer understanding of different types of bottom-up biases.

I will argue that there are at least three distinct types of bottom-up biases.

Feature-linking bias: bias that is due to which features an algorithm links among the features provided by the data on which it operates and how it links

[6] As I argue in the next section, feature-linking and fine-tunning biases are each cases of a biased network.

[7] In this respect, I disagree with Johnson, who argues that "bias is not a phenomenon one characterizes at the representation and algorithm level of analysis. Rather, bias is a computational-level phenomenon, which could be instantiated in multiple different ways at the underlying levels" (Johnson forthcoming-a, p. 14). Now Johnson does allow for representational bias, but argues that it is the computational level that unifies all the cases of bias. While I can see a case to be made that all cases of bias can be given a functional analysis (be they representational, neural, algorithmic, connectionist, or computational), I do not see that biases on, say, the neural or connectionist level can be given a unifying analysis on the computational level.

[8] For discussion, see [reference omitted for blind-refereeing].

	those features.
Fine-tuning bias:	bias that is due to how finely an algorithm discriminates or classifies with regard to one group compared to another group.
Training-sample bias:	bias that is due to the data on which the algorithm is trained.

While feature-linking bias is a special case of processing bias, training-sample bias is a special case of a biased input. Fine-tuning bias is due to amount of exposure to data either while an algorithm is trained or once it is used in the wild, so to speak. I will discuss each in detail.

5.1. Feature-linking bias

Once it has passed the test phase and is applied, an AI system operates on colossal and complex datasets. As in the training and test phases, these datasets consist of features. Exposed to new incoming data, the system expands and adapts the feature space created during the training and test phases. It picks up on regularities and correlations between features, links these features, and thereby creates an ever more complex feature space. The correlations may be so complex as to be beyond human understanding even by the programmers who produced the initial algorithms.

At every stage of developing this feature space, the algorithm projects that correlations encoded in the current feature space will hold in the future. The algorithm then processes future data within the framework of this feature space. So, it projects that the links held between features in the past will hold between features in the future.

A few examples will help illustrate this. In companies around the globe, algorithms are used to filter out promising job applications. These hiring algorithms can be biased in multiple ways. At Fox News, for example, an algorithm is used that considers a job application promising based on who was successful at Fox News in the past. It uses two criteria: that the employee stayed for five years and was promoted twice during that time. The algorithm then filters for applicants that are similar to the employees who satisfy those two criteria.

Now most Fox News employees who satisfy those two criteria have been white men. Unsurprisingly, most of the job applications that the algorithm deems worthy are applications of white men. The features these algorithms link can be as surprising as having played baseball in college. The bias is due to the algorithm linking features in problematic ways.

There are feature-linking biases where the stakes are even higher. One is the Northpointe recidivism risk software COMPAS (Correctional Offender Management Profiling for Alternative Sanctions). It is used in courtrooms across the United States. The software calculates a score predicting the likelihood of someone committing a crime in the future. Judges use these recidivism risk scores to

inform decisions about criminal sentencing.

As a propublica study revealed, the scores proved highly unreliable in predicting future crime: it was only marginally more reliable than a coin flip. Even more damning is that the algorithm generating the score is biased. It is much more likely to assume that African Americans are likely to re-offend. As a result, judges using this software are more likely to give African Americans harsh sentences.

What is the source of the bias? Northpointe's scores are based on answers to a set of questions. While race is not asked about directly, the questions provide a host of information about poverty, joblessness, and social marginalization. The software then links features that are proxies for race, like zip codes, in ways that leads to bias against African Americans.

Other examples of feature-linking bias are Google's online advertising algorithm. Ads for high-income jobs are shown to men more often than to women. Google searches for African American-sounding names yield ads for criminal background checks. Recently, it transpired that the algorithm used by Apple Card to determine the creditworthiness of applicants seems to set the credit line for men significantly higher than for women even when the women have a better credit score. In all these cases, the bias is due to the algorithm linking features in problematic ways. The features that are linked create bias.

It is important to note that the features linked may be proxy attributes, that is, seemingly innocuous attributes that correlate with socially sensitive attributes in the environment such that the former can serve (either purposely or accidentally) as proxies for the socially-sensitive attributes.[9] An example of a proxy attribute is a zip code that serves as a proxy for race in a racially segregated society. Northpointe's recidivism risk algorithm, for example, operates heavily on proxy attributes.

Feature-linking bias is a problem if the network tracks correlations between features in the environment that are not modally robust correlations but rather contingent correlations between features.[10] Data is noisy. The challenge for any data processing system—be it human or artificial—is to tease apart the signal from the noise and to avoid tracking noise. If the system tracks noise it can be subject to so-called overfitting. The algorithm overfits to the particularities of the data to which it has been exposed. In such cases the system tracks something highly correlated with the target rather than with the target.

A classic example of overfitting is an algorithm that was trained to distinguish wild wolves from domestic huskies. The algorithm was trained on images in which the wolves were predominantly pictured in a snowy landscape and the huskies in a grassy landscape. During this training phase, the

[9] For a discussion of proxy attributes, see Alexander 1992, p.167-173, Massey and Denton 1993, 51 ff, and Johnston forthcoming-a. See Adler et al (2016) for discussion of how to audit so-called 'black box' algorithms that appear to rely on proxy attributes in lieu of target attributes.

[10] For a development of the difference between statistically robust links and contingent links, see [reference omitted for blind-refereeing].

algorithm failed to learn the difference between wolves and huskies and instead learned that if there is snow it must be a wolf and if there is grass it must be a husky. The algorithm gave incorrect outputs roughly half the time.

Overfitting generates contingent links between features rather than modally robust links. Contingent links in AI systems lead to prediction errors and misrepresentation and the model will fail to generalize to new settings.[11]

5.2. Fine-Tuning Bias

What about the second type of bottom-up bias: fine-tuning bias? Fine-tuning bias is due to how finely an algorithm discriminates or classifies with regard to one group compared to another group. Such a bias is likely to be problematic if the relevant groups are demographic groups.

An algorithm exhibits fine-tuning bias if it is less finely tuned and so less informative and less reliable for features prevalent in certain parts of the populations. Different graphs and networks can make accurate predictions for the dominant group in a population but not for a minority or historically marginalized group if it is more finely tuned to features typical in one population but not the other. So a graph or network can work well for one group but not another.

Google's speech recognition algorithm is a good example. It performs better on male than on female voices. Why? It was trained disproportionately on the voices of men. As a consequence, the algorithm works best when decoding voices within the frequency of typical male voices. There are many similar cases. Automated facial recognition programs tend to identify women and people of color with lower accuracy than they identify white men. Here again the problem can be traced back to the skewed data on which the algorithms were trained: their training data consisted primarily of faces of white men. As a result, the algorithms do not master recognition of features more prevalent in faces of women and people of color. The algorithm discriminates more finely between features typical in people of one demographic group compared to features typical of people in other demographic groups. Thus, the bias is a fine-tuning bias.

Fine-tuning bias need not be due to the data on which the algorithm was trained. It can be due to data that the algorithm is exposed to after the training and test phases. An example is 23andme. It is much more fine-tuned to DNA of white people than that of other demographic groups, since white people use it significantly more.

Similar such fine-tuning biases occur in human perception. Consider Sasha who hears jazz for the first time. When listening to John Surman's recording of 'Doxology' for the first time, she will not discern much. As she becomes an expert, she will discern significantly more when listening to the very

[11] For related discussion of prediction errors and misrepresentation, see Hohwy 2013: 41-6, though Hohwy does not discuss the issue in the context of contingent links.

same recording. One explanation is that she develops more fine-grained perceptual capacities that allow her, for example, to discriminate between the sound of the trumpet and the sound of the piano even when they are playing at the same time and that allow her to hear differences between chords. In this sense, she develops the perceptual tools to discriminate more finely between the features of the incoming data.[12]

More fine-tuning is not always better. An example of a fine-tuning bias that is discriminatory due to being too fine-tuned for a marginalized group is the predictive policing algorithm PredPol. It is used by law enforcement agencies across the country to identify potential "hotspots" for crime. Its algorithm operates on data that consists of historical records of the frequency of criminal activity in particular areas. On the basis of this historical data, the algorithm makes predictions about where police should be dispatched in anticipation of new crimes. If historical policing practices have been shaped by discrimination, then we can expect bias. If police have, due to historical patterns of racism, tended to over-patrol predominantly black neighborhoods, then we can expect predictive software to continue to identify those neighborhoods as potential hotspots. It then dispatches police disproportionately to those areas, creating and collecting more data with which to continue the vicious cycle.

5.3. Training-Sample Bias

Training-sample bias occurs when the distribution of one's training data does not reflect the actual environment in which the machine learning model will be running. Training sample bias can lead to both feature-linking and fine-tuning bias. Google's speech recognition algorithm and automated facial recognition programs mentioned earlier are examples. Each of these cases of fine-tuning bias are due to training sample bias. The data on which the algorithms were trained did not reflect the demography on which the algorithm was deployed. However, both feature-linking and fine-tuning biases can develop after the training and test phase.

Some training sample biases emerge not from a training sample consisting predominantly of, say, one demographic group despite being used by all demographic groups, but rather from problematic data labeling. The data on which an algorithm is trained is labeled by a programmer. If the programmer labeling the data is biased, her bias might affects how she labels the data. She might, for example, harbor biases against women, believing them to be less qualified to successfully pursue STEM degrees, and rate a male applicant for graduate school higher than an equally-qualified female applicant. A machine learning algorithm trained on data labeled in this manner will replicate her bias. This example shows how top-down and bottom-up biases are often intertwined in complex ways.

Training-sample biases entail feature-linking bias. However, one can have feature-linking bias

[12] For a development of this view, see [reference omitted for blind-refereeing].

without training-sample bias. After all, an algorithm that was trained on a non-biased training-sample, can then in the deployment phase operate on data thereby developing feature-linking bias.

The general point here is that if there are problematic patterns encoded in the training data, then the predictions of the model generated by this training data will have those same problematic patterns. If the data used in the training phase is biased, then the generalizations encoded in the model will reflect those biases.

6. Coda

Biased algorithms are a problem since they lead to biased outcomes. By repeating our past practices, algorithms not only automate the status quo and perpetrate bias and injustice, they amplify the biases and injustices of our society.

In this way, bottom-up biases can create toxic feedback loops. To illustrate, let's go back to the ads one gets when one searches certain names. Once an AI system delivers ads in this manner it is no longer simply replicating the existing biases in our society. It is amplifying them. If an employer googles the name of a job applicant and that search prompts an ad suggesting the person had an arrest record, this may well negatively influence her hiring decisions, unconsciously and even consciously.

AI systems are promoted on the understanding that they are less prone to error and bias than humans. They are supposed to be unprejudiced and objective. But they are subject to top-down biases due to being designed, developed, and interpreted by humans. And they are prone to bottom-up biases of the very same kind that affect our visual and cognitive systems. In our imperfect world, biased algorithms generate harmful feedback loops and reinforce human prejudices.

7. References

Adler, P., Falk, C., et al. 2016: 'Auditing Black-Box Models for Indirect Influence'. *2016 IEEE 16th International Conference on Data Mining (ICDM)*, pp. 1–10.

Alexander, L. 1992: "What Makes Wrongful Discrimination Wrong? Biases, Preferences, Stereotypes, and Proxies". *University of Pennsylvania Law Review*, 141(1): 149–219.

Antony, L. 2016: "Bias: Friend or Foe?" In Brownstein, M. and Saul, J. (eds.), *Implicit Bias and Philosophy, Volume 1: Metaphysics and Epistemology*, pp. 157–90. Oxford: Oxford University Press.

Brooks. J.A. and Freeman, J.B. 2017: "Neuroimaging of Person Perception: A Social-Visual Interface". *Neuroscience Letters.* (special issue on Functional Imaging of the Emotional Brain), 693: 40-43.

Brownstein, M. and Saul, J (eds.) 2016, *Implicit Bias and Philosophy, Volume 1: Metaphysics and Epistemology*, Oxford: Oxford University Press.

Carruthers, Peter 2017: "Implicit versus Explicit Attitudes: Differing Manifestations of the Same Representational Structures?" *Review of Philosophy and Psychology*, 9(1): 51–72.

Combs, G. M. and Griffith, J. 2007" "An Examination of Interracial Contact: The Influence of Cross-Race Interpersonal Efficacy and Affect Regulation". *Human Resource Development Review*, 6 (3): 222–244.

Johnson, G. (forthcoming-a), "The Structure of Bias." *Mind.*

—— (forthcoming-b), "Algorithmic Bias: On the Implicit Biases of Social Technology." *Synthese.*

Hills, P. and Lewis, M. (2011). "The own-age face recognition bias in children and adults". *The Quarterly Journal of Experimental Psychology.* 64 (1): 17–23.

Hooker, Sara 2021: "Moving beyond 'algorithmic bias is a data problem'," Patterns, 2: 1-4.

Levy, Neil 2015: 'Neither Fish nor Fowl: Implicit Attitudes as Patchy Endorsements'. *Noûs*, 49(4): 800–23.

Mandelbaum, Eric 2015: "Attitude, Inference, Association: On the Propositional Structure of Implicit Bias". *Noûs*, 50(3): 1–30.

Massey, D., and Denton. N 1993: *American Apartheid: Segregation and the Making of the Underclass.* Cambridge, MA: Harvard University Press.